

Práctica No. 5 (Sorting)

Ej. 1. Baje el archivo `Sorting.java` de Moodle. Este archivo contiene varios métodos de ordenación. El algoritmo del MergeSort se encuentra en el archivo, sin embargo el método `merge` no está implementado. Implemente este método utilizando arreglos auxiliares.

Ej. 2. Agregue una versión random del quickSort a la clase `Sorting` descripta en el ejercicio anterior. En este caso en vez de tomar el primer elemento del arreglo como pivot, utilice un generador de números random para obtener una posición aleatoria.

Ej. 3. Implemente el radix sort en la clase `Sorting`.

Ej. 4. Implemente el counting sort en la clase `Sorting`. Puede suponer que solo se ordenarán enteros.

Ej. 5. Baje la clase `ArrayGenerator` del Moodle. Utilizar esta clase, que genera arreglos aleatorios de un tamaño dado, para comparar los métodos de ordenación de la clase `Sorting` de la siguiente forma.

- Compare los métodos de ordenación para arreglos con 1000 elementos.
- Compare los métodos de ordenación para arreglos con 10000 elementos.
- Compare los métodos de ordenación para arreglos con 100000 elementos.
- Compare los métodos de ordenación para arreglos con 1000000 elementos.

Ej. 6. Averigüar sobre el algoritmo de sorting llamado Shell Sorting e implementarlo en la clase `Sorting`. Cómo se comporta en la práctica este algoritmo?

Ej. 7. OpenDsa (Open Source Interactive Data Structure and Algorithm) es una versión electrónica de un libro con materiales para cursos con diferentes temas relacionados con Ciencias de la Computación, entre ellos, Estructuras de Datos y Algoritmos. Este libro interactivo es novedoso porque contiene una integración de texto e imágenes con visualizaciones/simulaciones interactivas. Acceda a una instancia del libro a través del siguiente link: <http://algoviz.org/OpenDSA/Books/CS3114> y resuelva las actividades planteadas en las diferentes secciones del capítulo 7.